FINEID - S4-2

Implementation Profile 2

for Organizational Usage

v 2.1A

Population Register Centre (VRK)
Certification Authority Services
P.O. Box 70
FIN-00581 Helsinki
Finland
http://www.fineid.fi

Authors

Name	Initials	Organization	E-mail
Antti Partanen	AP	VRK	antti.partanen@vrk.intermin.fi
Markku Sievänen	MaSi	Setec Oy	markku.sievanen@setec.com

Document history

Version	Date	Editor	Changes	Status
2.1A	21.10.2004	AP	Additional note on the encoding of EF.CIAInfo, value notation example of EF.CIAInfo corrected, minor editorial corrections	Accepted
2.1	15.3.2004	AP	Public edition	Accepted
1.2	11.6.2003	MaSi	First edition	Draft

Table of contents

1. Introduction

This document describes an implementation profile of the FINEID S1 specification version 2.1. This implementation profile is for organizational cards containing Qualified Certificates issued by Population Register Centre (VRK).

FINEID S4-1 is used as a base specification for all FINEID S4 -implementations. Therefore this document describes only the differences between S4-1 and S4-2.

1.1. About FINEID specifications in general

The FINEID specifications are publicly available documents describing how to implement a public key infrastructure (PKI) using smart cards.

There is a straight correlation between the FINEID specifications, ISO/IEC 7816-15, IETF RFC 3280 (PKIX Certificate and CRL profile), and the PKCS standards. FINEID S1 specifies the framework for the content of an Electronic ID card. FINEID S2 describes the content of certificates. FINEID S4-1 and S4-2 are profiling documents. These documents specify the file and directory format for storing security-related information in smart cards (security tokens). The corresponding documents are listed in the table below.

FINEID document	FINEID comments	Based on
FINEID S1	Framework for the Electronic ID application in the smart card	ISO/IEC 7816-4 and ISO/IEC 7816-8
FINEID S2	CA-model and content of certificates published and administrated by Population Register Centre (VRK)	IETF RFC 3280 and ETSI TS 101862 Qualified certificate profile
FINEID S4-1	Implementation profile 1 for Finnish Electronic ID Card	ISO/IEC 7816-15, PKCS#15 v1.1, FINEID S1 and FINEID S2
FINEID S4-2	Implementation profile 2 for Organizational Usage	FINEID S4-1
FINEID S5	Directory specification	IETF RFC 2256, LDAPv2 and LDAPv3

FINEID S4-1 contains an implementation profile specifying how the FINEID S1 specification should be put into practice in FINEID context. FINEID S4-1 is mainly based on ISO/IEC 7816-15. However, because of ISO/IEC 7816-15 doesn’t specify the free space management of the EID application, FINEID S4-1 uses EF(UnusedSpace) file defined in PKCS#15 v1.1 to solve this problem. Also the specification “Application Interface for smart cards used as Secure Signature Creation Devices, Part 1 – Basic requirements, CEN/ISSS, CWA 14890-1:2004(E), Version 1 Release 9 rev2” has influenced to this document.

Full names for the FINEID specifications are listed below:

- FINEID S1 - Electronic Identity Application, Version 2.1
- FINEID S2 – VRK (PRC) CA-model and certificate contents, v2.1
- FINEID S4-1 - Implementation Profile 1 for Finnish Electronic ID Card, Version 2.1A
- FINEID S4-2 - Implementation Profile 2 for Organizational Usage, Version 2.1A

- FINEID S5 – Directory Specification, v2.1

FINEID specifications are available at

- **http://www.fineid.fi**

The PKCS standards are available at

- **http://www.rsasecurity.com/rsalabs**

IETF PKIX documentation and other IETF RFC’s are available at

- **http://www.ietf.org/rfc**

2. FINEID S4-2

FINEID S1 specifies the content of an Electronic ID application. However, there are many options and features that are left for the EID application issuer to decide (number of private keys in the EID application, key lengths etc.). This document contains an implementation profile complementing those options.

It should be noticed that the FINEID S1, S4-1 and S4-2 documents specify the requirements for the EID application only. In a multiapplication smart card there may exist several other applications in addition to the EID application.

3. Differences between FINEID S4-1 and FINEID S4-2

In general, differences between FINEID S4-1 and S4-2 are minor and it should be easy to support both profiles in software products. Most differences are related to label names, file access conditions or file existence.

3.1. CA certificates

Two CA certificates (keylength 2048 bit) shall be stored into the FINEID application. These can be used as starting points of trust for the card holder.

Difference to FINEID S4-1: Different intermediate CA certificate.

Root CA Certificate label	Signed by
‘VRK Gov. Root CA’ (FIN) ‘VRK Gov. Root CA’ (ENG) ‘VRK Gov. Root CA’ (SWE)	Self-signed

Intermediate CA Certificate label	Signed by
‘ VRK CA for Qualified Certificates’ (FIN) ‘ VRK CA for Qualified Certificates’ (ENG) ‘ VRK CA for Qualified Certificates’ (SWE)	‘VRK Gov. Root CA’

The contents of Root and CA certificates are described in the FINEID S2 specification.

3.2. EF.CIAInfo

Description

The CIAInfo file contains generic information about the application as such and it's capabilities. This information includes the serial number, algorithms implemented etc.

Notice: the AlgorithmInfo type is coded according to the PKCS#15 v1.1, meaning that the algId field (named as objId in ISO/IEC 7816-15) is defined as optional, and therefore not included in this specification.

Difference to FINEID S4-1: Different application label.

The label shall be set according to the table below:

Application label
‘ FINEID’ (FIN)
‘ FINEID’ (ENG)
‘ FINEID’ (SWE)

Value notation example

```
 {
   version          v2,
   serialNumber     169752222515401241, -- an example
   manufacturerID   “VRK-FINEID”,
   label            “FINEID”,
   cardflags       {authRequired},
   supportedAlgorithms { -- SEQUENCE OF AlgorithmInfo
      {
         reference    ‘00´H,
         algorithm    ‘01’H, -- CKM_RSA_PKCS from PKCS #11
         parameters   NULL,
         supportedOperations{compute-signature, decipher}
      }
      {
         reference    ‘01´H,
         algorithm    ‘03’’H, -- CKM_RSA_X_509 from PKCS #11
         parameters   NULL,
         supportedOperations{compute-signature, decipher}
      }
      {
         reference    ‘02´H,
         algorithm    ‘06’H, -- CKM_SHA_1_RSA_PKCS from PKCS #11
         parameters   NULL,
         supportedOperations{compute-signature}
      }
      {
         reference    ‘03´H,
         algorithm    ‘00’H, -- CKM_RSA_PKCS_KEY_PAIR_GEN from PKCS #11
         parameters   NULL,
         supportedOperations{generate-key}
      }
   }
```

```
    preferredLanguage   “en”,
 }
```

3.3. EF.PrKD

Description

This transparent elementary file (Private Key Directory) contains general key attributes such as labels, intended usage, identifiers etc. When applicable, it contains cross-reference pointers to authentication objects used to protect access to the keys. It also contains the pointers to the keys themselves.

Difference to FINEID S4-1:CommonkeyAttributes.accessFlags for “auth. and encipherment key” set to value {sensitive, neverExtractable} to indicate possible use of key recovery .

Value notation example

```
 {
    privateRSAKey : {
       commonObjectAttributes { -- CommonObjectAttributes
          label “auth. and encipherment key”,
          flags {private},
          authID ‘01’H
          accessControlRules: { SEQUENCE of AccessControlRule
             { --- AccessControlRule
                 accessMode { execute }
                 authId ‘01’H
             }
          }
       },
       classAttributes { -- CommonKeyAttributes
          iD ‘45’H,
          usage {decipher, sign, keyDecipher},
          -- native by default true (HW RSA)
          accessFlags {sensitive, neverExtractable},
          keyReference ‘00’H
       },
       subClassAttributes { -- CommonPrivateKeyAttributes
          keyIdentifiers { -- SEQUENCE OF KeyIdentifier
             {
                idType  4, -- Subject public key hash
                idValue OCTET STRING :
                   ‘11223344556677889900112233445566778899 00’H
                   -- Faked value of SHA-1 hash
             }
          }
       },
       typeAttributes { -- PrivateRSAKeyAttributes
          value indirect : path : {
             path ‘4B01’H
          },
          modulusLength 1024,
       }
    },
    privateRSAKey : {
       commonObjectAttributes { -- CommonObjectAttributes
          label “signature key”,
          flags {private},
          authID ‘02’H,
          userConsent ‘01’H   -- user consent required for each
                                 private key operation !!!
          accessControlRules: { SEQUENCE of AccessControlRule
             { --- AccessControlRule
                 accessMode { execute }
                 authId ‘02’H
             }
          }
       },
       classAttributes { -- CommonKeyAttributes
          iD ‘46’H,
          usage {nonRepudiation},
          -- native by default true (HW RSA)
          accessFlags {sensitive, alwaysSensitive, neverExtractable,
             cardGenerated},
          keyReference ‘00’H
       },
       subClassAttributes { -- CommonPrivateKeyAttributes
          keyIdentifiers { -- SEQUENCE OF PKCS15KeyIdentifier
             {
                idType  4, -- Subject public key hash
                idValue OCTET STRING :
                   ‘1122334455667788990011223344556677889900’H
                   -- Faked value of SHA-1 hash
             }
          }
       },
       typeAttributes { -- PrivateRSAKeyAttributes
          value indirect : path : {
             path ‘3F0050164B02’H
          },
          modulusLength 1024,
       }
    }
 }
```

In DER encoding the outermost SEQUENCE OF is omitted.

The content of the actual private key files is completely card specific. Operations possible to perform with keys in these files may either be deduced by looking at the contents of the CIAInfo file or by external knowledge of the card in question.

3.4. EF.CD #1

Description

This transparent elementary file contains attributes and pointers to card holder certificates **'auth. and encipherment cert.'** (Certificate #1) and **'signature certificate'** (Certificate #2). Information in this file contains certificate attributes such as labels, key identifiers, pointers to certificate files etc.

Difference to FINEID S4-1: Access condition for update is CHV (PIN 1).

Access conditions

Access method	Access condition
Read	ALW
Update	CHV (PIN 1)

Value notation example

```
 {
    x509Certificate : {
       commonObjectAttributes { -- CommonObjectAttributes
          label "auth. and encipherment cert.",
          flags { modifiable }
          accessControlRules: { -- SEQUENCE OF AccessControlRule
             { --- AccessControlRule
                 accessMode { read }
                 NULL  --always
             }
             { --- AccessControlRule
                 accessmode { update }
                 authId '01'H
             }
          }
   },
       classAttributes { -- CommonCertificateAttributes
          iD '45'H
          -- By default authority FALSE i.e. not CA cert
          requestId { -- KeyIdentifier
             idType  3, -- Issuer and serial number hash
             idValue OCTET STRING :
                '1122334455667788990011223344556677889900'H
                -- Faked value of SHA-1 hash
          }
       },
       typeAttributes { -- X509CertificateAttributes
          value indirect : path : {
             path '3F004331'H
          }
       }
    },
    x509Certificate : {
       commonObjectAttributes { -- CommonObjectAttributes
          label "signature certificate",
          flags { modifiable }
          accessControlRules: { -- SEQUENCE OF AccessControlRule
             { --- AccessControlRule
                 accessMode { read }
                 NULL  --always
             }
             { --- AccessControlRule
                 accessmode { update }
                 authId ‘01’H
             }
          }
       },
       classAttributes { -- CommonCertificateAttributes
          iD ‘46’H
          -- By default authority FALSE i.e. not CA cert
          requestId { -- KeyIdentifier
             idType  3, -- Issuer and serial number hash
             idValue OCTET STRING :
                ‘11223344556677889900112233445566778899OO’H
                -- Faked value of SHA-1 hash
          }
       },
       typeAttributes { -- X509CertificateAttributes
          value indirect : path : {
             path ‘3F0050164332’H
          }
       }
    }
 }
```

In DER encoding the outermost `SEQUENCE OF` is omitted.

Files 3F00/4331 and 3F00/5016/4332 should contain DER-encoded certificate structure in accordance with ISO/IEC 9594-8.

3.5. Certificate #1

Description

This file contains the card holder’s **‘auth. and encipherment cert.’** containing the public key corresponding to the private RSA **‘auth. and encipherment key’** (Private RSA Key #1). The certificate in this file is DER encoded.

Difference to FINEID S4-1: Access condition for update is CHV (PIN 1).

Access conditions

Access method	**Access condition**
Read	ALW
Update	CHV (PIN 1)

3.6. Certificate #2

Description

This file contains the card holder's **'signature certificate'** containing the public key corresponding to the private RSA **'signature key'** (Private RSA Key #2). The certificate in this file is DER encoded.

Difference to FINEID S4-1: Access condition for update is CHV (PIN 1).

Access conditions

Access method	Access condition
Read	ALW
Update	CHV (PIN 1)

3.7. EF.CD #3 (trusted certs)

Description

This transparent elementary file contains attributes and pointers to trusted CA certificates. These certificates are used as a starting points of trust for the card holder (e.g. when verifying other certificates). Originally this file will contain pointers to the following CA certificates:

- o 'VRK Gov. Root CA' (self signed)
- o **'**VRK CA for Qualified Certificates**'** (signed by 'VRK Gov. Root CA')

Difference to FINEID S4-1: Different intermediate CA certificate.

Value notation example

```
 {
   x509Certificate : {
      commonObjectAttributes { -- CommonObjectAttributes
         label "VRK Gov. Root CA",
         flags {}
         accessControlRules: { SEQUENCE of AccessControlRule
            { --- AccessControlRule
                accessMode { read }
                NULL  --always
            }
         }
      },
      classAttributes { -- CommonCertificateAttributes
         iD '48'H
         authority TRUE, -- CA certificate
         requestId { -- KeyIdentifier
            idType  2, -- Subject key identifier
            idValue OCTET STRING :
               '11223344556677889900112233445566778899OO'H
               -- Faked value of subjectKeyIdentifier extension
            }
         },
         typeAttributes { -- X509CertificateAttributes
            value indirect : path : {
               path ‘3F004334’H
            }
         }
      }
      x509Certificate : {
         commonObjectAttributes { -- CommonObjectAttributes
            label “VRK CA for Qualified Certificates”
            flags {}
            accessControlRules: { SEQUENCE of AccessControlRule
               { --- AccessControlRule
                   accessMode { read }
                   NULL  --always
               }
            }
         },
         classAttributes { -- CommonCertificateAttributes
            iD ‘47’H
            authority TRUE, -- CA certificate
            requestId { -- KeyIdentifier
               idType  2, -- Subject key identifier
               idValue OCTET STRING :
                  ‘1122334455667788990011223344556677889900’H
                  -- Faked value of subjectKeyIdentifier extension
            }
         },
         typeAttributes { -- X509CertificateAttributes
            value indirect : path : {
               path ‘3F004333’H
            }
         }
      }
   }
```

In DER encoding the outermost `SEQUENCE OF` is omitted.

Files 3F00/4333 and 3F00/4334 should contain DER-encoded CA certificates in accordance with ISO/IEC 9594-8.

4. Certificates

The contents of the certificates are described in the FINEID S2 specification.

5. PUK-codes

Whole PUK-code is shipped with FINEID S4-2 card.